Relazione progetto QT Aprile 2020

Nome: Benito Francesco Antonio

Cognome: Duretto

Matricola: 829795

Email: b.duretto@campus.unimib.it

Introduzione

Il progetto consisteva nel realizzare un'applicazione con interfaccia grafica responsive che stampasse a video due liste di hyperlink di artisti(Universal e EMI) prese da due link differenti e alla fine mostrare tre grafici: uno che mostra il numero di elementi nelle due liste e gli altri due, uno per ogni lista, che mostrano il numero di artisti per iniziale.

Struttura dell'App

La classe QObject GetFileOnWeb ha i metodi per ricevere e salvare i dati da un determinato url.

La classe QThread FillThread ha i metodi per manipolare e gestire i dati salvati.

La classe QMainWindow ArtistiWindow è incaricata di istanziare le classi GetFileOnWeb e FillThread, per poi inserire nella gui i dati. Gestisce i signal dei vari oggetti istanziati per tenere un ben preciso loop di eventi.

GetFileOnWeb

Il loop si avvia tramite il metodo downloadFile, tramite QNetworkRequest e QNetworkReply, classi già implementate del framework, scarico i dati e li inserisco in una stringa da passare, poi, ai thread.

Quando termina invia un segnale ad ArtistiWindow

FillThread

Viene avviato da ArtistiWindow quando è possibile, e il lavoro del thread è quello di prendere la stringa ottenuta da GetFileOnWeb ed elaborarne il contenuto.

Per ogni link presente nella stringa, con il metodo giveLink viene realizzata una stringa "artista" passata ad ArtistiWindow insieme alla lettera iniziale dell'artista.

Quando termina invia un segnale ad ArtistiWindow.

ArtistiWindow

Ha una struttura dati custom per tenere traccia delle diverse iniziali degli artisti e il loro conteggio.

Il metodo starts() inizializza gli oggetti per ricevere i dati e i thread per la gestione di questi. Quando un oggetto GetFileOnWeb finisce, passa i dati ottenuti al thread FillThread corrispondente e quest'ultimo elabora i dati.

Ogni volta che un thread formatta un dato in una stringa, avvia il metodo corrispondente print_artisti_universal o print_artisti_emi per inserire l'elemento correttamente nella lista e conteggiare ed eventualmente inserire una nuova lettera da mettere nel grafico "numero di artisti per iniziale".

Quando il riempimento delle liste è concluso stampa i chart con i dati ottenuti, tramite il metodo create_chart_artisti.

Il Distruttore termina in modo sicuro i thread con quit().

Scelte implementative

Ho voluto usare i QThread, perché mi infastidiva il freezing dell'applicazione. Così l'utente nel frattempo può scorrere in tutta tranquillità le liste e interagire con gli hyperlink anche se non sono ancora del tutto riempite e i chart non ancora caricati.

Le coppie link-artisti con un link vuoto, vengono comunque conteggiate e mostrate nelle liste, ma il loro hyperlink porta ad una pagina 404.

Per la coppia link-artisti con gli url interi (es: https://xx.wikipedia.org/ecc..) viene estrapolata la nazione xx per avere un url consistente. Analizzando i file, ho notato che solo gli artisti che non sono presenti nella regione "en" hanno l'intero url e quindi ho optato per questa soluzione.

Le coppie link-artisti senza il nome di un artista, vengono automaticamente scartate per non creare dati inconsistenti e quindi non conteggiati nei grafici.

Non sapendo se effettivamente un underscore fosse, o meno, presente nel nome di un artista li ho trattati tutti come spazi.

Nei chart per il conteggio degli artisti per iniziale, la quantità di ogni inziale è indicata specificatamente nella legenda con "<iniziale> : <quantità>".
Nota: la legenda dei due chart per il conteggio degli artisti per iniziale, se risulta tagliata, è scorribile con il mouse.